

DATABASE Expert

Most people hate filling in forms, but luckily help is at hand. This month Kay Ewbank comes to the rescue of two readers who have problems creating forms in Access



FREE ORACLE

Oracle is to release a free version of its database. Rival companies have attracted new developers to their products with free entry-level versions. Oracle has released a beta of a product called Oracle Database 10g Express Edition, for 32-bit Windows and Linux systems. The software can be downloaded free from the Oracle website, and used for development as well as producing limited finished applications. You can only use the Express Edition with up to 4GB of data and 1GB of RAM, and only on one processor per server. The product is based on the same code base as Oracle's other 10g databases, but with some options removed.

ACCESS SERVICE PACK PROBLEM

If you're using Access to manage linked Excel data, be careful before installing the latest service pack. If you install the Microsoft Office 2003 Service Pack 2 or the update for Access 2002 dated 18th October 2005, you will no longer be able to change any linked data. Instead, you'll have read-only access to the spreadsheet data. The thinking behind it seems to be that changing linked data could quickly result in an unstable system, although this hasn't previously stopped developers using linked spreadsheets. More details about the problem are on the Microsoft Knowledge Base at <http://support.microsoft.com/kb/904953>.

Q I have developed a rather large database in Access that runs our small CCTV security company. This database handles everything from initial sales enquiries through to maintenance renewals and so on. It also has the engineer's diary (which I want to synchronise to Pocket PCs, but that's another subject).

One of the problems I have is that I would like the diary shown in an Outlook style or similar graphical way. I have tried various methods but I never seem to get it right. I tried using VBA automation to create an Exchange or Outlook entry. This worked well, and I use the same method to create automatic tasks, emails and reminders for deskbound staff.

But while I can create new Outlook items reliably, if the diary is later edited in Access, the two become out of sync. I want my graphic view of the diary to be totally up to date, even if it is read-only, as the engineer doesn't need to edit appointments or create them.

I'm using Access 2000 on the clients and the database was created with Access XP Developer.

James Wilson

A There are several ways to tackle this problem. One option would be to look for a third-party diary component; there are several available, and I'd be interested to hear from any readers who have experience of using any.

One that looks pretty reasonable and flexible – and which you can try before you buy – is Calendar Tools 3 from DBI Components (www.dbi-tech.com). This lets you display a form very similar to the Outlook diary view, with appointments taking up the correct amount of space across the diary. The only drawback is that if you decide to buy the components, they cost a couple of hundred pounds. You also need to allow a certain amount of time to

get to grips with any component, and Calendar Tools 3 is no exception.

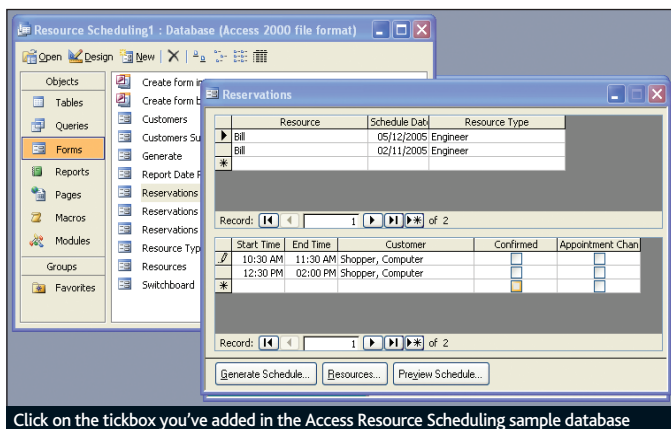
Another possibility that offers a solution with fairly minimal work is to base your form or report on the sample database template that comes with Access, called Resource Scheduling. This is designed to diarise access to resources such as meeting rooms, but your engineer can be treated as a resource as well. If you look at the way the tables in the Resource Scheduling database work, you'll see that you can enter bookings for individual resources, then view a resource schedule for those resources for particular dates. The layout may not be quite what you'd hoped for – it uses a standard report format – but you can at least see the diary organised on a day-at-a-time view.

However, as you have Outlook and you've got as far as creating diary entries, we'd be tempted to pursue this route. You say that if the diary entries are created in Access and are then later updated in Access, the versions become out of sync. Updating entries already entered in Outlook isn't easy, but it isn't impossible.

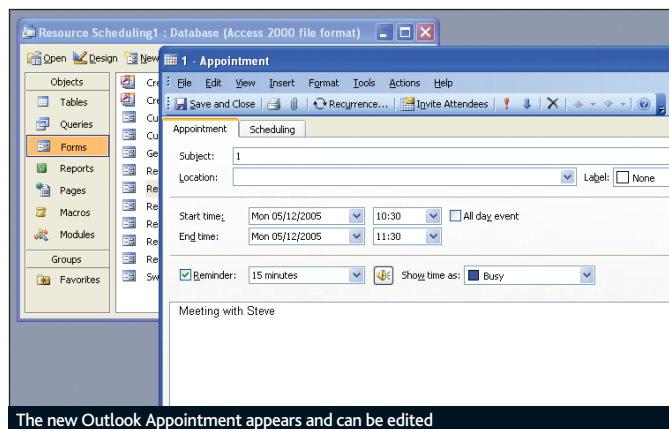
When you create a diary entry in Outlook, you can keep a note of its EntryID. Each item created in Outlook is given a unique EntryID, and you can use it to retrieve the item later. This means that if an appointment in Access is then changed, you can retrieve the corresponding entry in Outlook and change it to reflect the changes in your database.

The code below does this for the Resource Manager sample database in Access. An extra field has been added to the database's Schedule Details table to hold the EntryID, and we've called that field EntryID. My first procedure creates a new appointment – it should be attached to the Confirmed check box in the Resource Time subform of the sample database:

```
Private Sub Confirmed_Click()  
    Dim objOLApp As Outlook.Application
```



Click on the tickbox you've added in the Access Resource Scheduling sample database



The new Outlook Appointment appears and can be edited



```
Dim Newappt As Outlook.AppointmentItem
Set objOLApp = New Outlook.Application
Set Newappt = objOLApp.CreateItem(olAppointmentItem)
dttime = Forms!reservations.[reservations subform].Form!ScheduleDate
Newappt.Start = dttime + Me.[ScheduleStartTime]
Newappt.End = dttime + Me.[ScheduleEndTime]
Newappt.Subject = Me.[CustomerID]
Newappt.Display
Newappt.Save
Me.[entryid] = Newappt.entryid
End Sub
```

The first thing this routine does is create an Outlook Application object and an Outlook Appointment item. The Outlook Application object is then used either to open Outlook if it's closed, or make use of an already-open instance of Outlook. This works despite the fact that we only refer to a New Outlook Application, because Outlook can only be opened once at a time.

A new Appointment Item is created in the Newappt object, then we set up the items for the appointment from the details in the subforms. The Reservations form has two subforms. One holds the details of the times of the appointments and the other holds the details of the date, the customer and the resource you're managing. For simplicity and brevity, this subroutine assumes that the current Outlook diary is the correct one for our engineer resource.

The date of the appointment has to be retrieved from a different form to the Resource Time subform, to which our code is attached. This is done by the line:

```
dttime = Forms!reservations.[reservations subform].Form!ScheduleDate
```

This works because a subform is a property of a main form just like any other control. Next use:

```
Forms!myform.mysubform.Form
```

This gets you the form called mysubform on the main form called myform. If any of your names have spaces in them, you need to surround them by parentheses, as we've done with [reservations subform]. The actual control holding the date is called ScheduleDate.

This date value is stored in dttime, and has the start time and end time added to it separately, and those values are then copied into the Start and End properties of the appointment item.

We've set the subject of the message to be the CustomerID from the subform. It's not very informative, as it shows the ID rather than the name, but we didn't want to make the code any more complex. In a proper application you'd look up the customer name and store that along with some more identifying information.

Consider the section of the code:

```
Newappt.Display
Newappt.Save
```

```
Me.[entryid] = Newappt.entryid
```

This displays the new appointment item, so the user can add information of their own, and it is then saved. Saving generates the Entry ID and, finally, this is stored in a control on the form, which is bound to the EntryID column in the table.

We can create an appointment item that duly appears in Outlook. Now comes the task of retrieving it if the appointment is changed in Access. We've added

another check box to the form called ApptChanged. This is ticked when the appointment is changed. In a production system, you'd need to add extra code to untick the box after changes have been synchronised. You might well choose a different method such as adding a command button to the main form – again, we've stayed with the simplest alternative so as not to obscure the method.

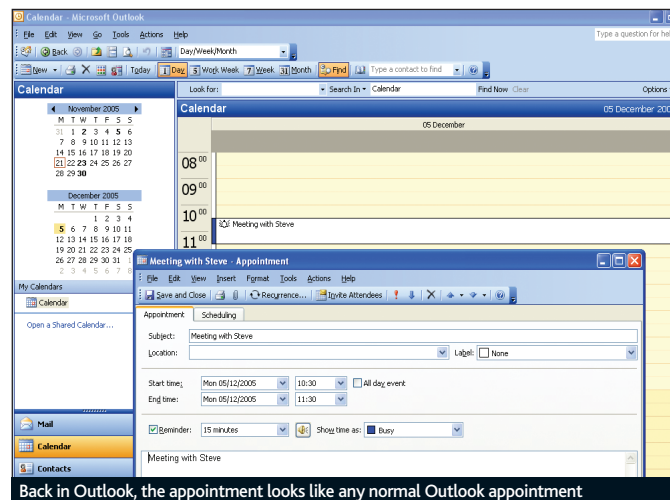
On the tick box clicked item, you need code along the following lines:

```
Private Sub ApptChanged_Click()
Dim objOLApp As Outlook.Application
Dim Chngappt As Outlook.AppointmentItem
Dim myNamespace As Outlook.Namespace
Dim myApts As Outlook.MAPIFolder

Set objOLApp = New Outlook.Application
Set myNamespace = objOLApp.GetNamespace("MAPI")
Set myApts = myNamespace.GetDefaultFolder(olFolderCalendar)
Set Chngappt = myNamespace.GetItemFromID(Me.[entryid])
dttime = Forms!reservations.[reservations subform].Form!ScheduleDate
Chngappt.Start = dttime + Me.[ScheduleStartTime]
Chngappt.End = dttime + Me.[ScheduleEndTime]
Chngappt.Subject = Me.[CustomerID]
Chngappt.Display
Chngappt.Save
End Sub
```

This has some similarities with the previous procedure, so that you create an Outlook application and set up an Outlook Appointment Item. To use an EntryID in VBA you need to work with Outlook's Namespace object. This root object can represent any data source, but the only data source that is supported in Outlook is MAPI. The MAPI namespace provides access to all the Outlook data that is stored in the user's mail stores, including the Appointment Items.

You can find any item in the NameSpace if you know its EntryID, and of course we do, because



Back in Outlook, the appointment looks like any normal Outlook appointment

we cunningly stored it when we created the item. Consider the line:

```
Set Chngappt = myNamespace.GetItemFromID(Me.[entryid])
```

This shows the appointment we created and stores it in Chngappt. We can now change the values in that appointment to reflect the changes that have been made in the Access form using the same techniques as the previous procedure.

Q I've written a crosstab query, but the criteria aren't working. If I specify a static criterion in the query such as 'January' for the month, it works fine. However, I want to let the user choose the month on a form. I have tried specifying the month they enter using an expression such as [Forms]![Calendar]![Month] but then I get an error message. The message says the Microsoft Jet database engine does not recognise [Forms]![Calendar]![Month] as a valid field name or expression. Where am I going wrong?

Trevor Stone

A Make sure that the form to which you're referring – Calendar in your case – is open when you're running the query. In addition, for a Crosstab query to work you need to write the parameter name and its type in the Parameter dialog box as well as on the criterion line.

To do this, open the query in Design View. From the Query menu, choose the Parameters option. A dialog box will appear. In the left-hand column, add the parameter you want to use. So this would be:

```
[Forms]![Calendar]![Month]
```

In the right-hand column, you need to specify the data type you expect the user to enter, so in your case this would be Text. You should then find that your query works as expected. **CS**

CONTACT

KAY EW BANK

Kay is Computer Shopper's database expert and has worked as a consultant and author for 20 years

kay@computershopper.co.uk